Software Cost Estimation Using Function Point Analysis

Anggi Yhurinda Perdana Putri[1], Apol Pribadi Subriadi[1]

Abstract*—Software Cost Estimation is an important activity in software development projects. Various measurement methods are widely applied to estimate software cost, and one of them is Function Point Method. Function Point (FPA) method measures the software by measuring software functionality that provided to users based on design logic. The measurement consist of several steps: calculate Unadjusted Function Point (UFP), Value Adjustment Factor (VAF), Adjusted Function Point (AFP) and estimate the software cost. Software project that became the object of this research were four licensing projects from local government with total cost 229.680.000. The estimated total cost of four projects using Function Point method is 216.956.881 rupiahs. The percentage gap between the actual with the function point that is equal to 5.54%. It can be concluded that FPA is an accurate method and this cost estimation can be used by developers in determining the cost of software to be built.

***Keywords*—Software Development, Software Cost Estimation, Function Point Analysis.**

I. INTRODUCTION

Software measurement is the most crucial activity in the software development process. One of the determinants of software project success is Cost Estimation and effort[1]. Many software projects fail because of time estimates, human resources and cost estimates[2]. In the process of developing software technology is changing very fast, so the rules of measurement must be adjusted with the changes. Therefore, there should be a general methodology that is independent of technology in the software industry. Function Point Analysis (FPA) fits perfectly into this category[3]. Function Point Method (FPA) is a method that quantifies the software functionality assigned to users based on logic design[4]. FPA was first introduced by Alan J. Albrecht from IBM in 1979[5]. The International Function Point User Group (IFPUG) officially declared that the Function Point method is appropriate for all types of software[6]. Based on Software Productivity Research, software measurement with Function Point can significantly increase the likelihood of success of on-time and on-budget software projects[7].

In the midst issue of electronic government (e-Government), which is one of the nonprofit business in software engineering, should be aware about estimated cost of software procurement projects, which inevitably leads to budget allocations. Eliminating or erroneously planning a software procurement project may result in less than maximum results[8]. In line with the results of the Standish Group survey[9], until 2015 about 71% of software development projects outline failure. So the objective of this study is to estimate the software cost especially in public service applications. The selection of 4 public service applications (Industrial Registration, Industrial Allowance, Principal Approval, and Certificate of Company License) is because the author has found a very little research on the implementation of FP methods in the realm of government.

[1]Anggi Yhurinda Perdana Putri and Apol Pribadi Subriadi are with Departement of Information Systems, Institut Teknologi Sepuluh Nopember, Surabaya, 60111, Indonesia. E-mail: anggiyhuri@gmail.com; apolpribadi@gmail.com.

II. FUNCTION POINT ANALYSIS

Function point is a method that breaks the system into smaller components so that it can be understood and analyzed better[10]. Function Point method is a method with approach oriented to the functionality and complexity in estimating the size of the software and further to estimate the effort and cost estimation for software development[1]. In the FPA procedure there are a variety of transactions, comprising the incoming and outgoing data to be processed on the system that consist of[11]:

1. External Input (EI). Process data or control information that comes from outside the application boundary. The processed data maintains one or more ILFs.
2. External Output (EO). Send data or control information outside the application boundary.
3. External Inquiries (EQ). Present information to a user through the retrieval of data or control information from an ILF or EIF.
4. Internal Logical Files (ILF). A user identifiable group of logically related data or control information maintained within the boundary of the application[6].
5. External Interface Files (EIF). A user identifiable group of logically related data or control information referenced by the application, but resides in the boundary of another application.

In the computation phase, each transaction is sorted by the amount of data that they use. Logical transaction or file sorted based on the entities (called RET or Referenced Entity Types) and attributes (called DET or Data Entity Types). Functional transaction sorted based on the attribute numbers (DET), which moved out of the line and the numbering of logical transaction references. Then the whole categorized (low, average, or high) which each category given value as the value of Function Point (FP)[12].

A. Function Point Counting Procedure.

Based on IFPUG, the following is the calculation steps of function point method. 1) Determine the type of Function Point count, 2) Identify the application boundary, 3) Identify EI, EO, EQ, ILF, EIF and their complexity, 4) Calculate Unadjusted Function Point Count 5) Determine the Value Adjustment Factor (VAF), 6) Calculate the final and 7) Adjusted Function Point Count (AFP).

III. FUNCTION POINT COUNT

A. Calculate Unadjusted Function Point Count.

Unadjusted Function Point is a weighting factor to each of EI, EO, EQ, ILF, and EIF based on its complexity. Weighting was the summed and become the number of UFP. Table 1 is an example of the counting process.

TABLE 1.
UNADJUSTED FUNCTION POINT COUNT

Component Types	Complexity Level						Total
	Low		Average		High		
	Weighting Factor	Point	Weighting Factor	Point	Weighting Factor	Point	
EI	(3) x 17	51	(4) x 3	12	(6) x 4	24	87
EO	(4) x 5	20	(5) x 1	5	(7) x 0	0	25
EQ	(3) x 5	15	(4) x 1	0	(6) x 1	6	21
ILF	(7) x 5	35	(10) x 2	20	(15) x 0	0	55
EIF	(5) x 2	10	(7) x 0	0	(10) x 0	0	10
Total Unadjusted Function Point							198

The complexity is based on the number of data element types (DETs) and file type referenced (FTRs). Rate the complexity of elementary processes (EI, EO, EQ) using the complexity matrix shown in Table 2 and 3.

TABLE 2.
DET AND FTR FOR EI

File Type Referenced (FTR)	Data Elements (DET)		
	1-4	5-15	> 15
< 2	Low (3)	Low (3)	Average (4)
2	Low (3)	Average (4)	High (6)
> 2	Average (4)	High (6)	High (6)

TABLE 3.
DET AND FTR FOR EO AND EQ

File Type Referenced (FTR)	Data Elements (DET)		
	1-5	6-19	> 19
< 2	Low (4)	Low (4)	Average (5)
2 or 3	Low (4)	Average (5)	High (7)
> 3	Average (5)	High (7)	High (7)

The functional complexity is based on the number of data element types (DETs) and record element types (RETs) associated with the ILF and EIF. Rate the complexity of data functions using the complexity matrix shown in Table 4 and 5.

TABLE 4 .
DET AND RET FOR ILF

Record Element Type (RET)	Data Elements (DET)		
	1-19	20-50	> 51
1	Low (7)	Low (7)	Average (10)
2 – 5	Low (7)	Average (10)	High (15)
> 6	Average (10)	High (15)	High (15)

TABLE 5.
DET AND RET FOR EIF

Record Element Type (RET)	Data Elements (DET)		
	1-19	20-50	> 51
1	Low (5)	Low (5)	Average (7)
2 – 5	Low (5)	Average (7)	High (10)
> 6	Average (7)	High (10)	High (10)

B. Determine the Value Adjustment Factor (VAF).

Value Adjustment Factor (VAF) is based on 14 general system characteristics (GSCs) that rate the general functionality of the application being counted (Table 6).

TABLE 6.
GENERAL SYSTEM CARACTERISTICS

General System Characteristic	Score	General System Characteristic	Score
Data Communication	5	On-line update	4
Distributed data processing	4	Complex processing	2
Performance	4	Reusability	5
Heavily used configuration	2	Installation ease	5
Transaction rate	1	Operational ease	3
On-line data entry	5	Multiple sites	5
End-user efficiency	5	Facilitate change	3
Total Degree Influence = 54			

0 – Not present or no influence
1 – Incidental influence
2 – Moderate influence
3 – Average influence
4 – Significant influence
5 – Strong influence throughout

Each characteristic has associated descriptions that help determine the total degree of influence (TDI) of that characteristic. The degree of influence for each characteristic range on a scale of zero to five – from no influence to strong influence. The VAF is given by

$$VAF = (Total\ Degree\ Influence \times 0.01) + 0.65$$
$$= (54 \times 0.01) + 0.65 = 1.19 \quad (1)$$

Function point count ± 35 percent to produce the adjusted function point. In this example set VAF =1.19 (it varies from project to project).

C. Calculate the final Adjusted Function Point Count (AFP).

The AFP determined by combining VAF and VAF value by using formula (2). The value of AFP identified as the size of application that counted. The AFP result is listed in Table 7. AFP given by

$$AFP = UFP \times VAF = 198 \times 1.19 = 235.62 \quad (2)$$

IV. Cost Estimation

A. Effort Value Project.

To change the AFP values that have been obtained into effort, the AFP value must be multiplied by the Effort Rate (ER) with man-hour units per FP. Equation (3) is used to obtain the effort.

$$Effort = AFP \times ER = 235.62 \times 8.2 = 1932.084 \quad (3)$$

Recent research conducted by Subriadi, et.al gives ER value of 8.2 man-hours for development software project of small and medium scale[13]. Table presents the result of effort per project.

TABLE 7.
EFFORT VALUE PER PROJECT

Project Name	Function Point	ER	Effort
Certificate of Industrial Registration	235.62	8.2	1932.084
Certificate of Industrial Business Permit	253.47	8.2	2078.454
Certificate of Approval	250.38	8.2	2053.116
Certificate of Company Registration	229.9	8.2	1885.18

B. Effort Value Activities

To identify the activities undertaken to complete the software development project. The activity required to complete the project depends on the software development model used. In the study used the list of activities and distribution effort provided by Primandari[14]. The next step is count the effort per activity which calculated by formula (4). Therefore, the list of effort per activity for the 4 projects is presented in Table 8

$$Effort\ Value\ per\ Activities = Effort\ Value\ per\ Project \times Activity\ distribution\ (\%) \quad (4)$$
$$= 1932.084 \times 1.17\%$$
$$= 22.60538$$

TABLE 8.
EFFORT VALUE PER ACTIVITIES

No	Activity	%	Project Name (Person-hours)			
			TDI	IUI	PP	TDP
	Software Development phase					
1	Requirements	1.17%	**22.60538**	24.31791	24.02146	22.056606
2	Specifications	6.75%	130.4157	140.2956	138.5853	127.24965
3	Design	5.57%	107.6171	115.7699	114.3586	105.00453
4	Implementation	55.65%	1075.205	1156.66	1142.559	1049.1027
5	Integration Testing	6.42%	124.0398	133.4367	131.81	121.02856
6	Acceptance & deployment	5.60%	108.1967	116.3934	114.9745	105.57008
	Ongoing activities & quality and testing					
7	Project management	2.55%	49.26814	53.00058	52.35446	48.07209
8	Configuration management	3.58%	69.16861	74.40865	73.50155	67.489444
9	Quality assurance	0.66%	12.75175	13.7178	13.55057	12.442188
10	Documentation	9.76%	188.5714	202.8571	200.3841	183.99357
11	Training & support	0.62%	11.97892	12.88641	12.72932	11.688116
12	Evaluation & testing	1.67%	32.2658	34.71018	34.28704	31.482506
	Total	100.00%	1932.084	2078.454	2053.116	1885.18

C. Costs per Activities.

The activity is determining pay rate per activities. Pay rate is obtained from a pay rate that applies in a particular country or region. In this study, we used pay rate released by Inkindo[15] for the index. Cost per activity is obtained by the formula (5). In Table 9 is the results of cost per activities that indicate the cost per project.

$$Cost\ per\ Activities = Pay\ Rate\ (index=1) \times (index\ per\ region) \times Effort\ Value\ per\ Activities \quad (5)$$
$$= 52.020 \times 1.02 \times 22.60538$$
$$= 1,199,451$$

TABLE 9.
COST PER ACTIVITIES

No	**Activities**	**Pay Rate (IDR/hour) index=1**	**Project ID (IDR)**			
			TDI	**IUI**	**PP**	**TDP**
		Index	**1.02**	**1.02**	**1.02**	**1.02**
		Software Development phase				
1	Requirements	52,020	1,199,451	1,290,318	1,274,588	1,170,332
2	Specifications	52,020	6,919,908	7,444,143	7,353,393	6,751,917
3	Design	52,020	5,710,205	6,142,797	6,067,911	5,571,582
4	Implementation	22,591	24,775,749	26,652,700	26,327,783	24,174,284
5	Integration Testing	19,322	2,444,631	2,629,830	2,597,770	2,385,284
6	Acceptance & deployment	19,322	2,132,388	2,293,933	2,265,968	2,080,622
		Ongoing activities & quality and testing				
7	Project management	63,910	3,211,701	3,455,012	3,412,893	3,133,733
8	Configuration management	52,020	3,670,114	3,948,153	3,900,022	3,581,017
9	Quality assurance	19,322	251,317	270,356	267,061	245,216
10	Documentation	8,620	1,657,995	1,783,601	1,761,857	1,617,745
11	Training & support	10,305	125,912	135,450	131,176	122,855
12	Evaluation & testing	19,322	635,909	684,084	675,744	620,471
Total			52,735,280	56,730,377	56,036,165	51,455,059

Pay rate in table 9 (Column number 3) is a pay rate with index = 1 based on Inkindo [14] is a pay rate applied in Jakarta. Since the project was developed in Surabaya, the index for East Java province is 1.02.

V. SUMMARY

Based on the research, the result of cost estimation using FPA compared with the actual cost is listed in Table 10.

TABLE 10.
COST ESTIMATION RESULT

No	**Nama Proyek**	**Actual Cost**	**Function Point**
1	Industrial Registration	44,300,000	52,735,280
2	Industrial Allowance	47,080,000	56,730,377
3	Principal Approval	46,800,000	56,036,165
4	Certificate of Company License	91,500,000	51,455,059
	Total	229.680.000	216.956.881

Software measurement is the most crucial activity in the software development process. In times of rapid changes in the global business scenario, there is a need to have industry accepted common methodology for software sizing that bind all professionals into common understanding in software business. Function Point (FPA) method measures the software by measuring software functionality that provided to users based on design logic. The measurement consist of several steps: calculate Unadjusted Function Point (UFP), Value Adjustment Factor (VAF), Adjusted Function Point (AFP) and estimate the software cost. Software project that became the object of this research were four licensing projects from local government with total cost 229.680.000, while using FPA = 216.956.881. Through the case study, the percentage gap between the actual compared with the function point that is equal to 5.54%. It can be concluded that FPA is an accurate method to measuring the size of the project as it is linked directly to system requirements and functionality. Since it is independent of technology and completely based on what end User receives, it will help establish an effective communication among different stakeholders of the software development project. This Research can be used by developers in determining the cost of software to be built especially in the field of public service applications.

REFERENCES

[1] S. Suharjito, A. Widodo, and B. Prasetyo, "Perancangan sistem estimasi biaya proyek pengembangan software," in *Seminar Nasional Aplikasi Teknologi Informasi (SNATI)*, 2006.

[2] N. Noerlina, "Perencanaan Manajemen Proyek Sistem Informasi dan Teknologi Informasi Online Bisnis," *J. Piranti War.*, vol. 11, no. 3, pp. 440–450, 2008.

[3] H. K. Raju and Y. T. Krishnegowda, "Software sizing and productivity with function points," *Lect. Notes Softw. Eng.*, vol. 1, no. 2, pp. 204–208, 2013.

[4] W. Pradani, "Kajian Metode Perhitungan Metrik Function-Point dan Penerapannya pada Dua Perangkat Lunak yang Dipilih," *J. Al-Azhar Indones.*, vol. 2, no. 1, pp. 28–34, Jun. 2013.

[5] A. J. Albrecht and J. E. Gaffney, "Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation," *IEEE Trans. Softw. Eng.*, vol. SE-9, no. 6, pp. 639–648, Nov. 1983.

[6] IFPUG, *Function Point Counting Practices Manual*. New Jersey: IFPUG, 2010.

[7] R. Georgi and T. Vogt, "Illustrative Example of a Function Point Analysis for the NASA Crew Exploration Vehicle Guidance, Navigation & Control Flight Software." Houston, 2008.

[8] R. S. Dewi, A. P. Subriadi, and Sholiq, "A modification

complexity factor in function points method for software cost estimation towards public service application," *Procedia Comput. Sci.*, vol. 124, pp. 415–422, Jan. 2017.

[9] S. Wojewoda and S. Hastie, "Standish Group 2015 Chaos Report - Q&A with Jennifer Lynch," *InfoQ: Software Development News, Videos & Books*, 2015. [Online]. Available: https://www.infoq.com/articles/standish-chaos-2015. [Accessed: 04-Jun-2017].

[10] D. Longstreet, "Function Points Analysis Training Course." Longstreet Consulting Inc., 2004.

[11] K. van den Berg, T. Dekkers, and R. Oudshoorn, "Functional Size Measurement applied to UML-based user requirements," in *Proceedings of the 2nd Software Measurement European Forum (SMEF2005)*, 2005, pp. 69–80.

[12] A. Živkovič, I. Rozman, and M. Heričko, "Automated software size estimation based on function points using UML models," *Inf. Softw. Technol.*, vol. 47, no. 13, pp. 881–890, Oct. 2005.

[13] A. Pribadi Subriadi, sholiq Sholiq, and P. Agustin Ningrum, "Critical review of the effort rate value in use case point method for estimating software development effort," *J. Theor. Appl. Inf. Technol.*, vol. 59, no. 3, pp. 735–744, 2014.

[14] P. L. Primandari A and S. Sholiq, "Effort distribution to estimate cost in small to medium software development project with use case points," *Procedia Comput. Sci.*, vol. 72, pp. 78–85, Jan. 2015.

[15] Ikatan Nasional Konsultan Indonesia (INKINDO), *Pedoman Standar Minimal Tahun 2013*. Jakarta: Ikatan Nasional Konsultan Indonesia (INKINDO), 2013.